

Apellido y Nombre: DNI:

1. **Práctica:** Se dispone de la información de los productos de un supermercado. De cada producto se tiene Código, Nombre, Rubro (1..20) y precio. Se pide implementar un programa que guarde en una estructura adecuada los productos de los rubros que tengan 10 productos.
2. Indique para las siguientes proposiciones, si son **Verdaderas o Falsas**. Justifique cada caso.
 - En la técnica de corrección de debugging es necesario analizar los casos límites del problema..
 - Un vector siempre se utiliza teniendo en cuenta la dimensión lógica.
 - Una función puede devolver un tipo de dato registro, real, booleano, integer, entre otros.
 - Un programa que utiliza sólo variables globales no requiere modularización.

3. Dada la siguiente declaración de tipos de datos y variables, justificar para cada sentencia numeradas son válidas o inválidas:

<pre> program ejercicio3; type cadena100 = string[100]; cliente = record codigo = integer; tel: integer; dir: cadena100; end; clientes = ^nodo; nodo = record datos: cliente; sig: clientes; end; var c: cliente; cli: clientes; </pre>	<pre> begin 1. read(c); 2. new(c); 3. read (cli); 4. c := nil; 5. cli := nil; 6. dispose(cli); 7. read(cli^.codigo); 8. write(c.codigo); end. </pre>
---	--

4. Describa las características de una estructura del tipo de dato vector y describa los pasos necesarios de la operación de búsqueda de un elemento en dicha estructura.
5. Teniendo en cuenta las referencias, calcule e indique la cantidad de **memoria estática** y el **tiempo de ejecución**. Muestre cómo obtiene resultado.

program ejercicio5;	Referencia	
<pre> type cadena20 = string[20]; notas = 2..10; alumno= record ape_nom: cadena20; nota: integer; end; vector = array [1..10] of ^alumno; var v: vector; i:integer; sum: integer; nota: notas; apeNom: cadena20; begin for i:= 1 to 10 do begin new(v[i]); read(nota); read (apeNom); v[i]^..nota:= nota; v[i]^..ape_nom:= apeNom; end; sum := 0; while (sum < 200) do begin read (nota); sum := sum + nota; end; end. </pre>	Char Integer Real Boolean String Puntero	1 byte 4 bytes 8 bytes 1 byte Longitud + 1 byte 4 bytes

Apellido y Nombre:

1. Indique qué imprime el siguiente código en Pascal.

```

program imprimir;
var
  a,b,c: integer;
procedure calcular (b: integer; var a: integer;)
begin
  c:= c + b; a:= (b + c) * 5; b:= (a + b) MOD 10;
  writeln(a,b,c);
end;
begin
  a:= 15; b:= 20; c:= b - a;
  calcular (a,c);
  writeln(a,b,c);
end.

```

2. Complete la columna de la izquierda con V o F según corresponda y justifique en **todos** los casos:

- | | |
|-------------------------------------|--|
| <input checked="" type="checkbox"/> | a. Las instrucciones dentro de una estructura de control for siempre se ejecutan al menos 1 vez. |
| <input checked="" type="checkbox"/> | b. Un programa modularizado es eficiente. |
| <input checked="" type="checkbox"/> | c. En el acceso a los campos de un registro es necesario respetar el orden en que fueron declarados. |
| <input checked="" type="checkbox"/> | d. Una variable global sólo puede ser accedida y modificada desde el cuerpo del programa principal. |
| <input checked="" type="checkbox"/> | e. Para utilizar una variable de tipo puntero p siempre se debe realizar new (p) . |
| <input checked="" type="checkbox"/> | f. Se pueden utilizar operaciones de entrada/salida sobre todos los campos de una variable de tipo registro. |
| <input checked="" type="checkbox"/> | g. La estructura de datos Lista es heterogénea. |
| <input checked="" type="checkbox"/> | h. Al asignar el valor de nil a un puntero se libera la memoria referenciada. |

3. Describa el tipo de dato vector y los pasos necesarios (pseudocódigo) para realizar la operación de insertar un vector en otro vector desde una posición determinada.

4. Realice el cálculo de la memoria estática y dinámica del siguiente programa.

Referencias: Integer (2), real (6), char (1), boolean (1) y puntero (4).

```

program ejercicio4;
const dimF = 10;
type
  cadenall= string[11];

  emple = record
    ape_nom: cadenall;
    edad: integer;
    sueldo: real;
  end;
vector = array[1..dimF] of ^emple;

```

```

var
  v: vector; e: emple; i: integer;
begin
  i:= 0;
  repeat
    i:= i + 1; new(v[i]);
    read(e.ape_nom, e.edad, e.sueldo);
    v[i]^:= e;
  until e.edad = 20;
  while (i > 0) do begin
    v[i]^ .sueldo:=v[i]^ .sueldo + v[i]^ .sueldo*0.25;
    i:= i - 1;
  end;
end.

```

5. Calcule el tiempo de ejecución para el programa del ejercicio 4).